

⑨ 日本国特許庁(JP)

⑩ 特許出願公開

⑫ 公開特許公報(A)

昭62-54328

⑤ Int. Cl.⁴

識別記号

庁内整理番号

④ 公開 昭和62年(1987)3月10日

G 06 F 7/22
15/20

7313-5B
8219-5B

審査請求 未請求 発明の数 1 (全5頁)

⑬ 発明の名称 ツリー構造作成編集装置

⑭ 特 願 昭60-194332

⑮ 出 願 昭60(1985)9月2日

⑯ 発 明 者 石 井 慎 一 郎 東京都港区芝5丁目33番1号 日本電気株式会社内

⑰ 出 願 人 日本電気株式会社 東京都港区芝5丁目33番1号

⑱ 代 理 人 弁理士 内 原 晋

PTO 2003-359

S.T.I.C. Translations Branch

明 細 書

発明の名称 ツリー構造作成編集装置

特許請求の範囲

データをツリー構造のノードとして記憶すると共に、前記ノードの親・兄弟・子を示す関連情報を記憶する意味情報記憶装置と、前記意味情報記憶装置に対してノード及び関連情報からなる意味情報の追加、削除をすると共に、前記意味情報を用いてツリー構造の作成編集を行い、前記意味情報を図形情報に変換する情報処理装置と、前記図形情報を表示する図形表示装置とを備えたことを特徴とするツリー構造作成編集装置。

発明の詳細な説明

(産業上の利用分野)

本発明は、ソフトウェア開発におけるシステム分析のような、上位概念を詳細化し、下位概念に展開し階層化していく概念詳細化設計に係り、特にツリー構造に図形化されて表示される情報で意

味的に継りを持つ情報の表示に好適なツリー構造作成編集装置に関する。

(従来の技術)

第3図は、本発明の図形表示装置3に表示される図形イメージであるが、従来の図形編集表示装置では、単に「販売管理」や「販売計画」の表示情報、例えば座標などの情報により表示していた。つまり、ノード間の親・兄弟・子の関連情報を記憶する手段及び関連情報を図表情報に変換する手段を備えていなかった。

(発明が解決しようとする問題点)

従来技術では、第3図の「販売管理」が「販売計画」に対して親の関係であるという情報は、図形編集表示装置により表示されている場合、利用者が記憶しておくか、または別の装置を用いて理解しなければならなかった。つまり、第3図で「販売管理」と「販売計画」の間のアークは単に図形表示上線であるという意味しか持っていなかった。

本発明の目的は、計算機を用いて図形を編集・表示する装置において、特に図形をツリー構造に限

り、単にツリー構造の作成・編集・表示するだけでなくノード間の親・兄弟・子という関連情報をも併せてアークとして表示出来、またノードの情報及び前記関連情報からなる意味情報を記憶する機能により前記問題点を解決しようとするものである。

(問題点を解決するための手段)

データをツリー構造のノードとして記憶すると共に、前記ノードの親・兄弟・子を示す関連情報を記憶する意味情報記憶装置と、前記意味情報記憶装置に対してノード及び関連情報からなる意味情報の追加、削除をすると共に前記意味情報を用いてツリー構造の作成編集を行い、前記意味情報を図形情報に変換する情報処理装置と、前記図形情報を表示する図形表示装置とを備えた事を特徴とするツリー構造作成編集装置が得られる。

(実施例)

以下、本発明を図面を用いて説明する。第1図は本発明に係るシステムの全体構成を示したものである。第1図中、1はツリー構造中の各データ(例えばシステム機能分析における各機能名として販売

きにも使用される。また、同図(b)に示すデータ関連部ではデータ間の関連を前記IDによる表形式で記憶している。例えば、ID1の子供関連にあるデータの中で最も自分に近いものはID2のデータであり、またID2のデータにとって親にあたるデータはID1のデータである。図(b)の内容を情報処理装置2を用いて、図形表示装置3上に表示したものが第3図である。同図では各データがツリー構造のノードとして表示されている。

情報処理装置2からわたされる図形情報を図形表示装置3で第3図のような図として表示するまでの流れを第8図を用いて説明する。まず、情報処理装置2のノードID認識手段4より、図形情報として処理対象ノードIDと、親ノードIDとがわたされ、それぞれ対象処理ノード認識処理11、親ノードID認識処理13で認識される。次にノード表示処理12で対象処理ノードが適当な位置に表示され、またアーク表示処理14で親ノード対象処理間にアークが表示される。

管理、販売計画などがある)及び、そのデータ間の親・兄弟・子に該当する関連情報を記憶する意味情報記憶装置を示す。2は意味情報記憶装置1中のデータをそのデータの識別ラベルIDを用いてノードID認識手段4によって図形情報に変換したり、入力されるコマンドに従ってノード作成手段5、ノード削除手段6、アーク編集手段7によりツリー構造におけるそれぞれの処理を行い図形情報に変換したり、終了コマンドによって意味情報記憶装置1にデータを格納する情報処理装置を示す。3は前記情報処理装置2で処理・変換された図形情報を表示する図形情報表示装置を示す。

次に、本発明の動作について例を用いて説明する。

第2図(a),(b)は意味情報記憶装置1の中に格納されている情報の一例を示す図である。システム分析における機能の一覧をツリー構造にまとめたときの記憶情報である。同図(a)に示すデータ部では各データに識別ラベルとしてIDをつけて記憶する。前記IDは、そのデータの表示の処理を行うと

このツリー構造編集装置で扱う処理として次の3つが挙げられる。

■ノード作成処理

ノードをふやす機能である。例えば、第4図のように第3図に示すツリー構造の中にデータとして「実績管理」を追加する処理である。その処理の流れを第4図、第5図(a),(b)の例と第9図のフローチャートを用いて説明する。

まず、作成コマンド入力処理21で作成コマンドをうけつけ、処理対象ノードID認識処理22によって第5図(a)の「実績管理」のノードID4が認識され、親ノード認識処理23によって第5図(a)の「販売計画」のノードID2が認識されて、共に図形情報に変換されて、図形表示装置3にわたされる。

続いて兄弟ノード認識処理24で、第5図(a)の「顧客別販売計画」のノードID5が認識される。次に、親ノードID1の子供のところのIDは第5図(b)では2であり、前記処理対象ノードID4よりも小さいので、この部分は変更しない。もし、逆ならば、親ノードIDの子供の部分に処理対象ノードIDが入ること

になる。この処理は第9図中25の判断及び26の処理で行われる。続いて、同図の27の判断、28の処理において「実績管理」が作成されたことによって、第5図(b)のID3の兄弟の部分に「実績管理」のID4が改めて代入される。以上の処理により、意味情報記憶装置2の内容が第2図より第5図に変化し、図形表示装置で表示される図形も第3図より第4図に変化し、ノード作成処理は終了する。

ノード削除処理

ノードを削除する機能である。例えば、第3図のように第4図に示すツリー構造の中の「実績管理」を削除する処理である。その処理の流れを第10図のフローチャートを用いて説明する。まず削除コマンド入力処理31で削除コマンドを受け付け、処理対象ノードID認識処理32で、削除対象ノードIDが認識され判断33、処理34でもし親ノードの子として登録されている場合はその部分を消去し、判断35、処理36で他ノードの兄弟として登録されている場合はその部分を消去し、続いて処理37で、対象ノードを消去する。以上の処理によ

り、意味情報記憶装置2の内容が、第5図より第2図に変化し、図形表示装置で表示される図形も第4図から第3図に変化し、ノード削除処理は終了する。

アーク編集処理

アークを付け替えることによって、ノード間の関連情報を変更する機能である。例えば第6図のように「顧客別販売計画」のデータの親を第3図における「販売計画」から「販売管理」に変更する処理である。処理の流れを第11図のフローチャートを用いて説明する。

まず、アーク編集コマンド入力処理41でアーク編集コマンドを受け付け、処理対象ノードID認識処理42によって、アーク編集対象ノードが認識され続いて新規ノード認識処理43で、新しく親となったノードIDが認識され、共に図形情報に変換されて図形表示装置3にわたされる。そして、親がかわったので、新しく兄弟となるノードのIDの認識が兄弟ノードID認識処理44で行われる。次に判断45、処理46でもし旧親ノードの子供に処理対象ノードIDが登録されているならば消去し、判断

47、処理48で他ノードの兄弟に処理対象ノードIDが登録されているならば消去し、以降、第9図における25、27の判断、26、28の処理と同様に第11図49、51判断50、52処理を行う。以上の処理により、意味情報記憶装置2の内容が第2図から第7図(a)、(b)のように変化し、図形表示装置で表示される図形も第3図より第6図に変化し、アーク編集処理は終了する。以上の処理を用いて、利用者にノード及び親・兄弟・子の関連情報からなる意味情報を図形情報として提供でき、また第2、5、7図のような形態で意味情報記憶装置2上に記憶できる。

(発明の効果)

本発明によれば概念の階層化において、階層化される概念のデータ間の関連情報に基いたツリー構造の作成・編集を行えるエディタを提供することができるので、利用者は関連情報を事前に理解しておく必要がなくなるという効果がある。

図面の簡単な説明

第1図は本発明のシステムの基本構成図、第2図(a)、(b)、第5図(a)、(b)、第7図(a)、(b)は本発明の意味

情報記憶装置に蓄積される情報の構造についての例に基いて記した例を示す図、第3、4、6図は本発明の図形表示装置上の画面イメージである。

また第8、9、10、11図はそれぞれ表示、ノード作成、ノード削除、アーク編集の各処理に対するフローチャートを示す図である。

図において、

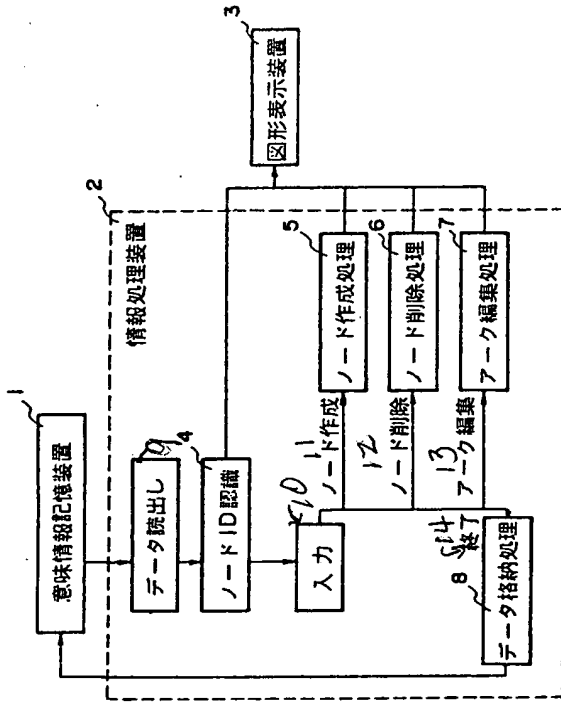
- 1は意味情報記憶装置、2は情報処理装置、
 - 3は図形表示装置、4はノードID認識処理、
 - 5はノード作成処理、6はノード削除処理、
 - 7はアーク編集処理、8はデータ格納処理
- それぞれ示す。

代理人 弁理士 内 原



図2

図1



(a)

データ部 5

ID	データ
1	販売管理
2	販売計画
3	商品別販売計画
4	顧客別販売計画

(b)

自ID	親ID	兄弟ID	子ID
1	0	0	2
2	1	0	3
3	2	4	0
4	2	0	0

図3

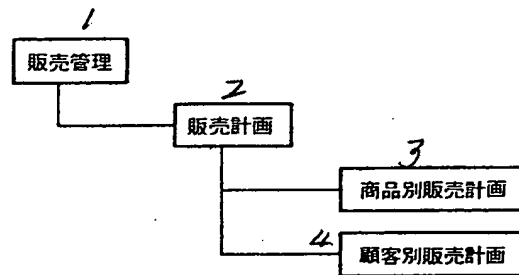


図4

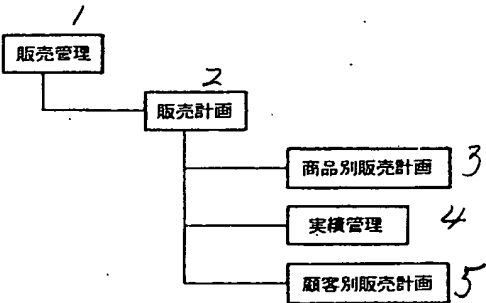


図6

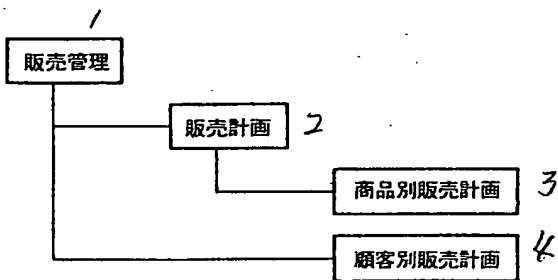


図5

(a)

データ部 6

ID	データ
1	販売管理
2	販売計画
3	商品別販売計画
4	実績管理
5	顧客別販売計画

(b)

自ID	親ID	兄弟ID	子ID
1	0	0	2
2	1	0	3
3	2	4	0
4	2	5	0
5	2	0	0

図7

(a)

データ部 5

ID	データ
1	販売管理
2	販売計画
3	商品別販売計画
4	顧客別販売計画

(b)

自ID	親ID	兄弟ID	子ID
1	0	0	2
2	1	4	3
3	2	0	0
4	1	0	0

図 8

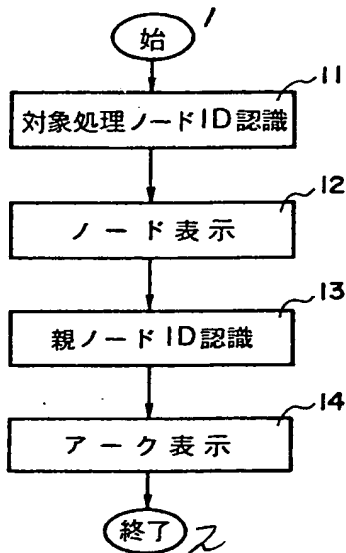


図 9

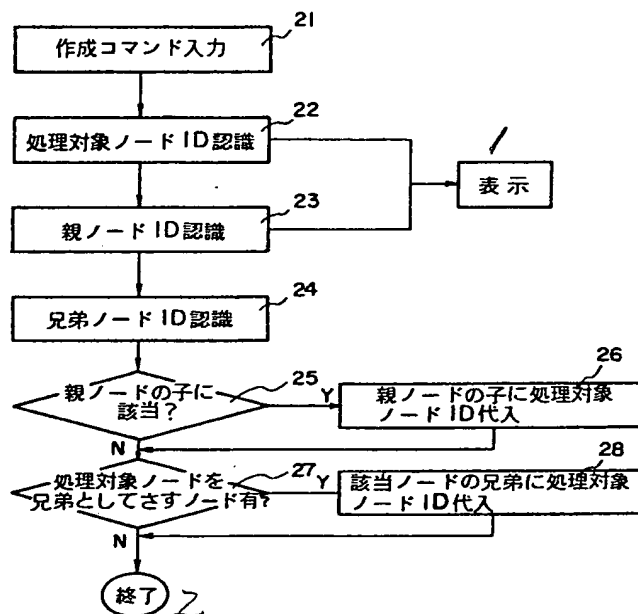


図 10

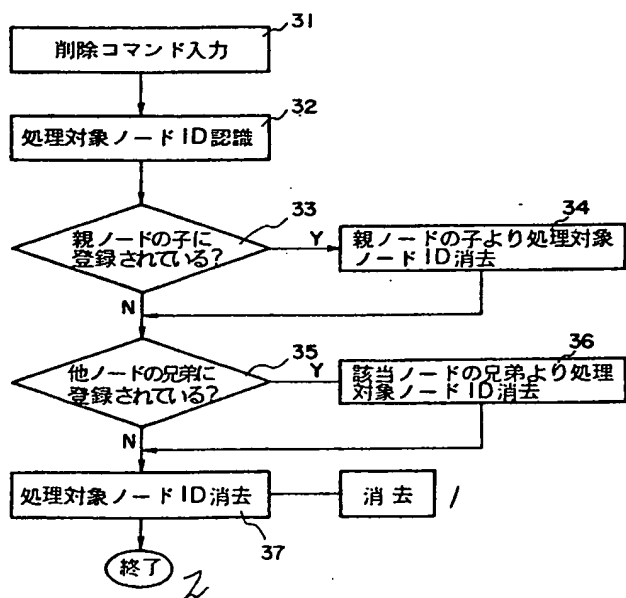
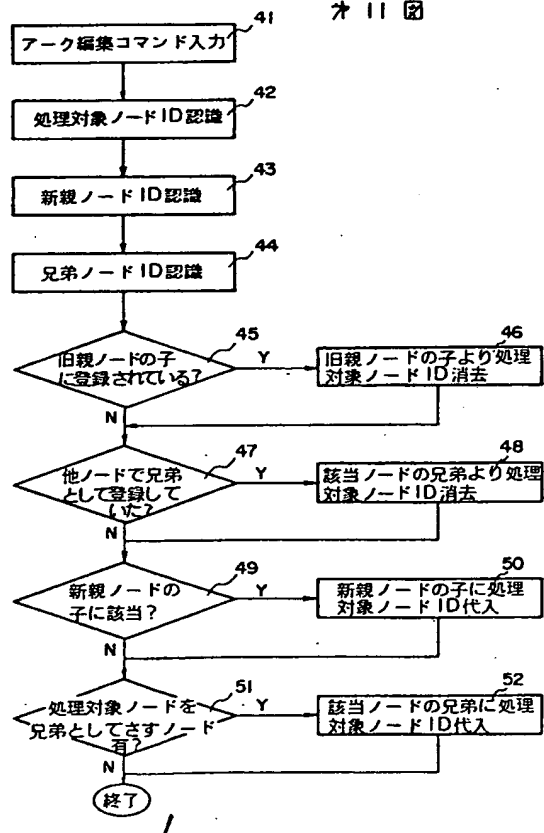


図 11



PTO: 2003-359

Japanese Published Unexamined Patent Application (A) No. 62-54328, published March 10, 1987; Application Filing No. 60-194332, filed September 2, 1985; Inventor(s): Shinichiro Ishii; Assignee: Nippon Electric Corporation; Japanese Title: Tree Structure Creating and Editing Devices

TREE STRUCTURE CREATING AND EDITING DEVICES

CLAIM(S)

A tree structure creating and editing device characterized by its comprising: a meaningful data memory for storing the data in form of nodes in a tree structure and for storing the relationship data indicating a parent, brother, and child of said node; a data processor that adds and deletes the meaningful data consisting of node data and of relationship data to and from said meaningful data memory, that creates and edits the tree structure by using said meaningful data, and that converts said meaningful data into graphical data; a graphic image display device that displays said graphic image data.

DETAILED DESCRIPTION OF THE INVENTION

(Field of Industrial Application)

The present invention pertains to a concept detailing structure like a system analysis in software development, wherein an upper order concept is broken down into lower order concepts in hierarchy, particularly to a tree structure creating and editing device appropriate for displaying the graphical data having relationship

among one another in form of a tree structure.

(Prior Art)

Fig. 3 shows a graphic image displayed on the graphic image display device 3 of the present invention. With the graphical image editing and displaying device of the prior art, it displayed only the “sales contro” and “sales plan” by using coordinates. Because it had neither a means to store the data indicating the relationship among the nodes, parent, brother and child, nor did it had a means to convert said relationship data into graphical data.

(Problems of the Prior Art to Be Solved)

In the prior art, when the relationship data that the “sales control” is a parent to the “sales plan” is displayed on the graphical image editing and displaying device, the relationship data had to be memorized by the user or had to be interpreted by using a separate device. In other words, the relationship between the “sales control” and “sales plan” did not mean anything more than merely a graphic delineation in Fig. 3.

The objective of the present invention is to present a computerized graphic image editing and displaying device, wherein the image is formed into a tree structure; it not only creates, edits, and displays the tree structure but can display the relationship data of parent, brother, and child in form of hierarchy. By so doing, the aforementioned problems can be solved by a function that stores the

meaningful data consisting of node data and relationship data.

(Means to Solve the Problems)

The tree structure creating and editing device of the present invention is characterized by its comprising: a meaningful data memory for storing the data in form of nodes in a tree structure and for storing the relationship data indicating the relationship among a parent node, a brother node, and a child node; a data processor that adds and deletes the meaningful data consisting of node data and of relationship data to and from said meaningful data memory, that creates and edits the tree structure by using said meaningful data, and that converts said meaningful data into graphical data; a graphic image display device that displays said graphic image data.

(Embodiment Example)

The present invention is explained below with reference to the drawings. Fig. 1 shows a configuration of the system of the present invention. In Fig. 1, 1 indicates the meaningful data memory device for storing each data (e.g., sales control and sales plan as the function name in the system function analysis) in the tree structure and the relationship data corresponding to the parent, brother, and child relationship among the data; 2 indicates the data processor that converts the data in the meaningful data memory device 1 into the graphic image data by node ID recognition means 4 by using the data discrimination label ID, that performs each

process in the tree structure according to the input command by node creation means 5, node deletion means 6, and arc editing means 7 and converts them into a tree structure, and that stores the data in the meaningful data memory 1 by the end command; 3 indicates the graphic image data display device for displaying the graphic image data processed/converted by said data processor 2.

The operation of the device of the present invention is explained below with reference to its example.

Fig. 2 (a)(b) show one example of the data stored in the meaningful data memory 1. This is the stored data when a list of functions in system analysis was put into a tree structure.

In the data section of Fig. 2 (a), an ID as a discrimination label is assigned to each data to be stored. This ID is used when the data is displayed. In the relationship data section of Fig. 2 (b), the relationship between the data is stored in form of said ID. For example, among the data having child relationship to ID1, the closest data to itself is ID2. The data which is a parent to the data ID2 is the data ID1. Fig. 3 shows the content of Fig. 2 (b) displayed on the graphic image display device 3 by using the data processor 2. In this graph, each data is displayed in form of node in a tree structure.

Fig. 8 explains the steps of process flow for displaying the image data handed from the data processor 2 in form of the graph of Fig. 3 by the image data display

device 3. First, the ID recognition means 4 of the data processor 2 hands the node ID to be processed and the parent node ID, which are each respectively recognized in the node recognizing/processing step 11 and in the parent node ID recognizing/processing step 13. Subsequently, the node to be processed is displayed at a proper position in node displaying/processing step. Also, in the hierarchy [”arc” is used in the Japanese text.] displaying/processing step 14, the hierarchy is displayed between the parent node processing steps.

The following three steps should be noted as the processes performed by the tree structure editing device.

*** Node creation process**

This process functions to add a node. For example, as shown in Fig. 4, the data of “performance control” is added to the tree structure of Fig. 3. This process flow is illustrated with reference to Fig. 4, Fig. 5 (a)(b), and to flowchart of Fig. 9.

First, the creation command is received in the creation command input process step 21; the node ID4 of “performance control” of Fig. 5 (a) is recognized in the node ID recognizing/processing step 22; the node ID 2 of “sales plan” in Fig. 5 (a) is recognized in the parent node recognizing/processing section 23 and both are converted into the graphic image data to be passed down to the graphic image display device 3.

Subsequently, node ID 5 of “sales plan by customer group” in Fig. 5 (a) is

recognized. Then, the ID of the child to the parent node ID1 is 2 in Fig. 5 (b), which is a smaller number the node ID4 to be processed, so this portion is not changed. If this situation is reverse, the node ID to be processed will be put in the child of the parent node ID. This is performed in the judgement step 25 of Fig. 9 and in the processing step 26 of Fig. 9. Subsequently, in the judgement step 27 of the same figure and in the processing step 28 of the same figure, the “performance control” is created; thus, the “performance control” ID4 is plugged into the brother ID3 of Fig. 5 (b). By the aforementioned process flow, the content of the meaningful data memory 2 is changed from that of Fig. 2 into that of Fig. 5, and the graphic image displayed on the graphic image display device is also changed from that of Fig. 3 into that of Fig. 4. The node creating process ends here.

*** Node deletion process**

This process functions to delete a node. For example, as shown in Fig. 3, “performance control” in the tree structure shown in Fig. 4 is deleted. The process flow for this deletion is illustrated with reference to the flowchart of Fig. 10. First, deletion command is received in the deletion command input processing step 31. The node ID to be deleted is recognized by the process node ID recognizing/processing section 32. If this node to be deleted is registered as a child to the parent node in the judgement step 33 and in the processing step 34, this node is deleted. If it is registered as a brother to another node in the judgement step 35 and in

the processing step 36, it is deleted, so the node to be deleted is deleted in the processing step 37. By the aforementioned process, the content of the meaningful data memory 2 is changed from that of Fig. 5 into that of Fig. 2, and the graphic image displayed on the graphic image display device also is changed from that of Fig. 4 into that of Fig. 3. The node deleting process ends here.

*** Hierarchy editing process**

This functions to change the relationship data of the nodes by replacing the hierarchy. For example, the parent to the data “sales plan customer group” is changed from the “sales plan” of Fig. 3 to “sales control.” The process flow for this is explained below with reference to Fig. 11.

First, the hierarchy editing command is received in the hierarchy editing command input processing step 41. In the process node ID recognizing/processing step 42, the node whose hierarchy is to be edited is recognized. Then, in new node recognizing/processing step 43, the node ID which became a new parent is recognized; and both are converted into the graphic data to be passed down to the graphic display device 3. Since the parent was replaced, the node ID which became a brother is recognized in the node ID recognizing/processing section 44. If the node to be processed is registered as the node ID as the child to the former parent in the judgement step 45 and in the processing step 46, this node is deleted. If the node ID to be processed is registered as the brother to another node in the judgement step 47

and in the processing step 48, this node is deleted. Hereinafter, the node is processed in judgement steps 49 and 51 and in processing steps 50 and 52 of Fig. 11 in the same way as in the judgement sections 25 and 27 and processing sections 26 and 28 of Fig. 9. By the aforementioned process, the content of the meaningful data memory 2 is changed from that of Fig. 2 to that of Fig. 7 (a) (b), and the graphic image displayed on the graphic image display device is also changed from that of Fig. 3 to that of Fig. 6. Thus, the hierarchy editing process ends here. By using the process explained above, the meaningful data consisting of nodes and relationship data of parent, brother, and child can be presented to the user in form of graphic image data, and they can be stored in the meaningful data memory 2 in the format shown in Fig. 2, Fig. 5, and Fig. 7.

(Advantage)

According to the present invention, by providing a hierarchy to the concepts, an editor that creates and edits the tree structure constructed on the bases of the hierarchical relationship data of multiple data can be presented, so the user does not need to know the relationship data in advance, which is advantageous.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a schematic diagram of the basic structure of the system of the present invention. Fig. 2 (a), (b), Fig. 5 (a), (b), and Fig. 7 (a),(b) show one example of the data structure stored in the meaningful data memory of the present invention.

Fig. 3, Fig. 4, and Fig. 6 show the image on the screen of the graphic image display device.

Fig. 8, Fig. 9, Fig. 10, and Fig. 11 show the flowchart of the display, node creation, node deletion, and arc editing processes.

- 1. Meaningful data memory**
- 2. Data processor**
- 3. Graphic display device**
- 4. Node ID recognition process step**
- 5. Node creation process step**
- 6. Node deletion process step**
- 7. Arc editing process**
- 8. Data storing process step**

Key to the figures

Fig. 1

- 1. Meaningful data memory**
- 2. Data processor**
- 3. Graphic image display device**
- 4. Node ID recognition**
- 5. Node creation process**

6. Node deletion process

7. Arc editing process

8. Data storing process

9. Data output

10. Input

11. Node creation

12. Node deletion

13. Arc editing

14. End

Fig. 2 (a)

1. Sales control

2. Sales plan

3. Sales plan by product group

4. Sales plan by customer group

Fig. 2 (b)

1. Data relationship section

2. ID of self

3. ID of parent

4. ID of brother

5. ID of child

Fig. 3

- 1. Sales control**
- 2. Sales plan**
- 3. Sales plan by product group**
- 4. Sales plan for customer group**

Fig. 4

- 1. Sales control**
- 2. Sales plan**
- 3. Sales plan by product group**
- 4. Performance control**
- 5. Sales plan by customer group**

Fig. 6

- 1. Sales control**
- 2. Sales plan**
- 3. Sales plan by product group**
- 4. Sales plan by each customer group**

Fig. 7 (a)

- 1. Sales control**
- 2. Sales plan**
- 3. Sales plan by product group**

4. Sales plan by customer group

Fig. 7 (b)

1. Data relationship section

2. ID of self

3. ID of parent

4. ID of brother

5. ID of child

Fig. 8

1. Start

11. Recognizing the node ID to be processed.

12. Node display

13. Recognizing parent node ID.

14. Arc display

2. End

Fig. 9

1. Display

21. Creation command input

22. Recognizing the node ID to be processed.

23. Recognizing the parent node ID.

24. Recognizing the brother node ID.

25. Is the node a child to the parent node?
26. The child of the parent node is replaced with the node ID to be processed.
27. Is there any node that claims the node to be processed as a brother?
28. The brother of the respective node is replaced with the node ID to be processed.
2. End

Fig. 10

31. Input of the deletion command.
32. Recognizing the node ID to be processed.
33. Is the node registered as a child to the parent node?
34. The child to the parent node is an object to be processed, so its node ID is deleted.
35. Is this node registered as a brother to the other node?
36. Node ID of the brother to the respective node is an object to be processed, so its node ID is deleted.
37. Node ID to be processed is deleted.

1. Deletion
2. End

Fig. 11

41. Input of the arc editing command
42. Recognizing the node ID to be processed.

43. Recognizing a new node ID.
44. Recognizing the brother node ID.
45. Is the node registered as a child to the former parent?
46. The child to the former parent is to be processed, so its node ID is deleted.
47. Is the node registered as a brother to other node?
48. The brother to the respective node is to be processed, so its node ID is deleted.
49. Is the node a child to a new parent?
50. The child to the parent is replaced with the node ID to be processed.
51. Is there any node that claims the node to be processed as its brother?
52. The brother to the respective node is replaced with the node ID to be processed.
1. End

Translations
U.S. Patent and Trademark Office
11/05/02
Akiko Smith